Computers and Mathematics with Applications 57 (2009) 476–482

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Cryptanalysis and improvement on a cryptosystem based on a chaotic map

Xingyuan Wang*, Canghai Yu**

Department of Electronic and Information Engineering, Dalian University of Technology, Dalian 116024, China

ARTICLE INFO

Article history:
Received 31 March 2008
Received in revised form 4 September 2008
Accepted 10 September 2008

Keywords:
Cryptanalysis
Encryption
Decryption
Remedial improvement
Logistic Map

ABSTRACT

In this paper, two fatal flaws of the cryptosystem which are based on the logistic map and proposed by Y. Wang and T. Xiang are pointed out. According to this, cryptanalysts could recover the plaintext by the chosen plaintext attacked in a short time. Therefore, the authors propose a remedial improvement which can avoid the flaws and enhance the security of the cryptosystem.

0. Introduction

A chaotic dynamics system has some good features such as: sensitive dependence on initial conditions; ergodicity; cycle tending to infinity. Under the control of certain parameters, we can obtain very good pseudo-random numbers. Therefore, chaotic cryptography has aroused extensive concern, and many scholars have proposed their chaotic cryptosystem [1–7] in recent years, such as Baptista's chaotic cryptosystem which is based on ergodicity [2], and Wang and Xiang's block cryptosystems [3,4], which are based on a Logistic Map. However, there are still many problems in the practical application of a chaotic cryptosystem, because some of the cryptographic algorithms take a long computing time and the length of the ciphertext is likely to be several times longer than that of plaintext; and some of the cryptographic algorithms cannot assure good security, which leads to a wide range of attacks. After successfully attacking the block cryptosystem based on Logistic Map proposed by Wang and Xiang [3,4], this paper presents a modified block cryptographic algorithm, and the study shows that the improved algorithm is more secure than the original algorithm.

1. Brief introduction of T. Xiang and Y. Wang's block chaotic cryptosystem

The key is the most important thing in block cryptosystems. Once the key is cracked, the cryptosystem will break down.

The cryptosystem proposed by T. Xiang and Y. Wang has some good features, such as fast encrypting and decrypting, short key, and no statistical characteristic. The size of encrypted file is close to the original one. This cryptosystem uses the logistic map

$$\tau(x) = \mu x(1-x), \quad x \in [0, 1], \mu \in [3.5699456, 4] \tag{1}$$

* Corresponding author. Tel.: +86 0411 84707827.
** Corresponding author. Tel.: +86 015904257219.
E-mail addresses: wangxy@dlut.edu.cn (X. Wang), Canghai.yu@gmail.com (C. Yu).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.09.042

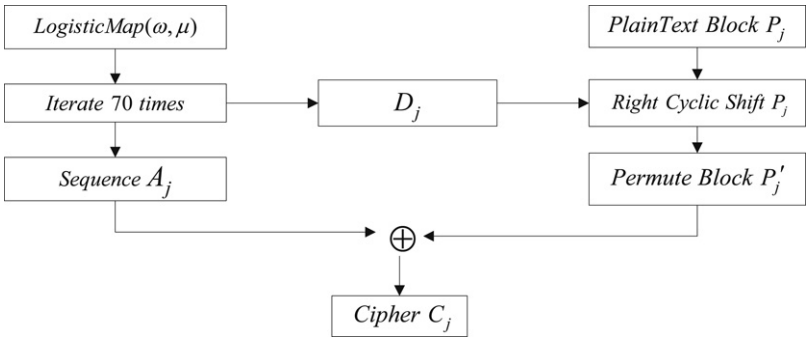

Fig. 1. Block diagram of the scheme.

to generate pseudo-random number. When $x \in [0, 1]$, $\mu \in [3.5699456, 4]$, Eq. (1) is a chaotic dynamic system; Eq. (1) sensitively depends on initial condition x_0, and the iterative value $\{x_i\}$ of Eq. (1) is close to random. Expressing the value of x in binary representation, we have

$$x_i = b_1(x)b_2(x)b_3(x)\ldots b_j(x)\ldots, x \in [0, 1], b_j(x) \in \{0, 1\}. \tag{2}$$

As a result, a binary sequence $B_j^n = \{b_j(\tau^n(x))\}_{n=0}^{\infty}$, where n is the length of the sequence and $\tau^n(x)$ is the nth iteration of the logistic map, can be obtained. The sequence is formed by all the jth bit of the x of the binary representation. Fig. 1 shows how to encrypt data using the cryptosystem.

(1) Get the start point ω which denotes the real value of x from the last N_0 transient time iterations, i.e., $\omega = \tau^{N_0}(x_0)$.

(2) Divide the message *PlainText* into subsequences with length of L bytes, (in this paper using $L = 8$)

$$PlainText = \underbrace{p_0 p_1 \cdots p_{L-1}}_{\omega_0} \underbrace{p_L \cdots p_{2L-1}}_{\omega_1} p_{2L} \cdots. \tag{3}$$

Let p_j denote the value of jth byte. Get 8 bytes of plaintext $p_j p_{j+1} \cdots p_{j+7}$ and combine them to form a binary message block P_j.

(3) Based on the method to generate binary sequences by iterating the logistic map 70 times, obtain the binary sequences $B_i^1 B_i^2 B_i^3 \cdots B_i^{64} B_i^{65} \cdots B_i^{69} B_i^{70}$ formed by all the third bits, i.e., $i = 3$ in Eq. (2). Divide the sequence into 2 parts, one is the front 64 bits $A_j = B_i^1 B_i^2 B_i^3 \cdots B_i^{64}$ and the other is $A_j' = B_i^{65} \cdots B_i^{69} B_i^{70}$. Convert A_j' into an integer D_j which is less than 64. This value will be used to iterate the logistic map successively for D_j times after the current block has been encrypted. Now, we get the key A_j and D_j.

(4) Permute the message block P_j with left cycles shift D_j bits to obtain a new message block P_j'.

(5) Perform the following manipulation with the sequences P_j' and A_j:

$$C_j = P_j' \oplus A_j; \tag{4}$$

where $\oplus$ denotes the XOR operation. As a result, the ciphertext block C_j for the message block P_j is obtained. Dividing the ciphertext block C_j into 8-bit partitions, we obtain the ciphertext $c_j c_{j+1} \cdots c_{j+7}$ of the plaintext bytes $p_j p_{j+1} \cdots p_{j+1}$, respectively. After C_j has been generated, we introduce two equations:

$$f(C_j) = c_j + c_{j+1} + \cdots + c_{j+7}; \tag{5}$$

$$D^* = D_j + f(C_j) \bmod 64. \tag{6}$$

(6) If all the plaintexts have been encrypted, the encryption process is finished. Otherwise, let $\omega = \tau^{D^*+70}$ and go to step 2.

The decryption process is almost the same as the encryption one. We only need to replace Eq. (4) with:

$$P_j' = C_j \oplus A_j. \tag{7}$$

After that, permute the block P_j' with right cyclic shift D_j bits to obtain the plaintext. Following the Eqs. (5) and (6), we can obtain the D^*. Iterate the logistic map with D^* times. Repeat these procedures until all the ciphertext has been decrypted.

2. The cryptanalysis and the attack of this algorithm

An attack of a cryptosystem means that cryptanalyst can obtain the plaintext without the key. In general, there are two ways to attack the traditional cryptosystem: ① the brute-force method. A cryptanalyst constructs a vector space in the binary sequence whose length is equal to the key to find out every possible key. If the length of the key is n, the capacity of the space is 2^n. ② the cryptanalysis method. A cryptanalyst can obtain the key or plaintext by analyzing the characteristic of

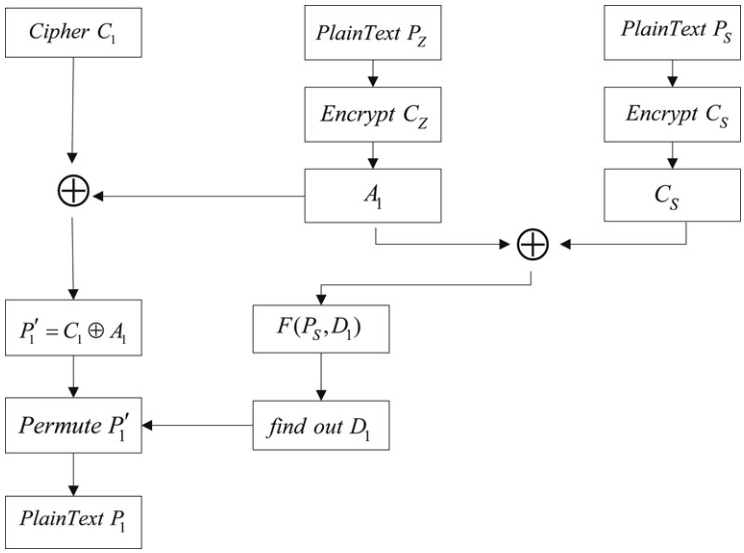

Fig. 2. Recover the first block of the ciphertext.

the cryptosystem. This method relies on the cryptographic algorithms and some pairs of the plaintext and ciphertext. This paper can obtain the keys by analysis of the cryptographic algorithms.

Traditionally, an attack by a cryptanalyst consists of [8,9]:

(1) Ciphertext only. The cryptanalyst has access only to a collection of ciphertexts or codetexts. In this case, the information is at a minimum. If poor information can deduce the plaintext, the cryptographic algorithm is fairly insecure.

(2) Known-ciphertext. The known-plaintext attack is an attack model for cryptanalysis where the attacker has samples of both the plaintext and its encrypted version (ciphertext) and is at liberty to make use of them to reveal further secret information; typically this is the secret key.

(3) Chosen plaintext. A chosen-plaintext attack is an attack model for cryptanalysis which presumes that the attacker has the capability in choosing arbitrary plaintexts to be encrypted and obtaining the corresponding ciphertexts. The goal of the attack is to gain some further information which reduces the security of the encryption scheme. In the worst case, a chosen-plaintext attack could reveal the scheme's secret key.

(4) Chosen ciphertext. A chosen-ciphertext attack is an attack model for cryptanalysis in which the cryptanalyst chooses a ciphertext and causes it to be decrypted with an unknown key.

Any cryptosystem which cannot resist the above attack is insecure.

The secret key of [3,4] actually is (μ, x_0). The difference between [3,4] is that [3] does not consist of Eqs. (5) and (6). That is the fatal flaw which leads to the same keystream if the secret keys are the same. The same keystream is easy to obtain plaintext by analyzing the pairs of plaintext and ciphertext. Ref. [4] has proposed a scheme to remedy the fatal flaw, trying to make the keystream depend on both the key and plaintext. This scheme adds Eqs. (5) and (6) into step 5. After using this scheme, the keystream is dependent on key and plaintext except the first key. Based on this flaw, this paper proposes two attack methods, which can obtain the keystream and plaintext.

Compared with the traditional block cryptosystem, the cryptosystem proposed by [4] eliminates statistics characteristic of the logistic map and uses a simple algorithm. Attacking this chaotic cryptosystem via a common brute-force exhaustive method costs a large amount of time and space. Hence, this paper chooses a cryptanalysis method to attack the scheme which is proposed in [4]. Our chosen ciphertext attack is described below and illustrated in Figs. 2 and 3.

(1) Choose a special plaintext block P_Z composed of all zeros, whose length is 64, and C_Z is the corresponding ciphertext. C_Z is fairly important, because C_Z is the first key of the keystream. We have

$$C_Z = F(P_Z, D_1) \oplus A_1, \tag{8}$$

where function $F(A, B)$ indicates permuting A by a left cyclic shift B bits. As P_Z is composed of all zeros, $F(P_Z, D_1)$ is equal to P_Z. Thus we have

$$C_Z = F(P_Z, D_1) \oplus A_1 = 0 \oplus A_1 = A_1. \tag{9}$$

(2) Choose another special plaintext block P_S, whose length is 64. P_S is

$$P_S = \underbrace{0111\cdots111}_{63},$$

in binary representation. The first bit in P_S is zero, and the rest are all one. C_S is the ciphertext which is corresponding to P_S. We have

$$C_S = F(P_S, D_1) \oplus A_1, \tag{10}$$

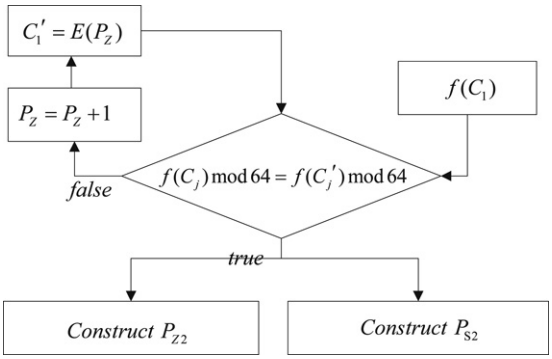

Fig. 3. Recover the second block of the ciphertext.

and the A_1 is already obtained. We get

$$C_S \oplus C_Z = F(P_S, D_1) \oplus A_1 \oplus A_1 = F(P_S, D_1). \tag{11}$$

There is only one zero in the $F(P_S, D_1)$ and we can find out its position easily. We know the D_1 by analyzing the $F(P_S, D_1)$ and P_S.

(3) Permute the following manipulation. C_1 is the first block of the ciphertext.

$$P_1' = C_1 \oplus A_1. \tag{12}$$

After that we obtain the first block of the plaintext by permuting the P' with right cyclic shift D_1 bits.

(4) Compute the value of $f(C_1)$ by using Eq. (5). Whatever the plaintext is, if the secret keys are the same, D_1 are the same.

$$f(C_1) \bmod 64 = f(C_1') \bmod 64. \tag{13}$$

Construct a block $P_{D_1^*}$ whose length is equal to P_1's, and C_1' is the corresponding ciphertext to $P_{D_1^*}$. Make sure the value $f(C_1') \bmod 64$ is equal to $f(C_1) \bmod 64$ by increasing the value of $P_{D_1^*}$. Actually we can obtain the same D^* by constructing $P_{D_1^*}$ in order to synchronize the iterating times in a logistic map.

(5) Construct a new sequence P_{Z2}, whose length is 64 more than $P_{D_1^*}$. The front of the P_{Z2} is filled with $P_{D_1^*}$ and the last 64 bits are filled with P_Z. Encrypt the P_{Z2}, and find out the last block of the ciphertext. That is the key A_2.

(6) Construct another new sequence P_{S2}, whose length is 64 more than $P_{D_1^*}$. The front of the P_{S2} is filled with $P_{D_1^*}$ and the last 64 bits are filled with

$$P_S = \underbrace{0111 \cdots 111}_{63}.$$

Encrypt the P_{S2}, and find out the last block of the ciphertext, C_{S2}.

(7) Permute the C_{S2} and A_2 with XOR operation which is similar to step 3. After analyzing the result, we can obtain D_2, then we can recover the P_2, the second block in plaintext. And then alter the value of the last block of P_{Z2} by increasing, until the Eq. (14) is satisfied.

$$f(C_2) \bmod 64 = f(C_2') \bmod 64, \tag{14}$$

if all the ciphertexts are recovered, algorithm will quit; if not, construct 2 new sequences whose lengths are 64 more than before and go to step 5, till all the ciphertexts are recovered.

We have another improved algorithm by using the recovered block instead of the conflicting block. The first 3 steps are the same as with algorithm 1.

(4) Obtain plaintext block P_1 via A_1, D_1.

(5) Construct a new sequence P_{Z2}, whose length is 64 more than P_1. The front of the P_{Z2} is filled with P_1 and the last 64 bits are filled with P_Z. Encrypting the P_{Z2}, find out the last block of the ciphertext. That is the key A_2.

(6) Construct another new sequence P_{S2}, whose length is 64 more than P_1. The front of the P_{S2} is filled with P_1 and the last 64 bits are filled with

$$P_S = \underbrace{0111 \cdots 111}_{63}$$

in Fig. 4. Encrypting the P_{S2}, find out the last block of the ciphertext, C_{S2}.

(7) We can obtain a plaintext block P_2 via A_2 and C_{S2_2}. And then construct 2 new sequences whose length is 64 more than before. The last 64 bits of the two blocks are filled with P_Z and P_S individually. The fronts of the 2 new sequences are filled with the entire recovered plaintext block. Repeating the procedure above, we can obtain the plaintext block P_3.

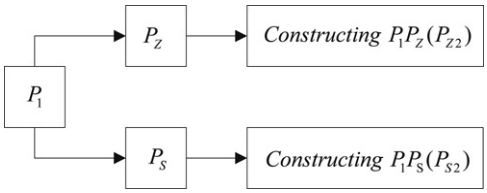

Fig. 4. Construct 2 new sequences.

Table 1
The first 2 blocks of the encryption of the plaintext "abcdefghijklnmopqrstuvwxyz" using $(\mu, x_0) = (3.999995, 0.17777)$.

j	P_j	D_j	A_j	C_j
1	0×6162636465666768	0×1f	0×5d55a4bc5a8ae20f	0×97996a6c984e24c7
2	0×696a6b6c6d6e6f70	0×29	0×1a53363e031eb2b3	0×ac658109bb2a0786
3	–	–	–	–

Table 2
Encryption of the plaintext P_Z and P_S with modified cryptosystem using $(\mu, x_0) = (3.999995, 0.17777)$.

	PlainText	D_j	A_j	*Cipher*
P_Z	0×0000000000000000	0×1f	0×5d55a4bc5a8ae20f	0×5d55a4bc5a8ae20f
P_S	0×7fffffffffffffff	0×1f	0×5d55a4bc5a8ae20f	0×a2aa5b42a5751df0

Table 3
Trying $f(E(P_{D_1^*}))$ using a brute-force method.

$P_{D_1^*}$	$f(E(P_{D_1^*}))$ mod 64
0×0000000000000000	0×00000027
0×0100000000000000	0×00000025
0×0200000000000000	0×0000002b
–	–
0×0800000000000000	0×00000017

If you want to recover the C_j, you have to recover the C_{j-1}. Construct two new sequences whose length is j blocks. The first $j-1$ blocks of the two sequences are filled with the entire recovered plaintext block. The last 64 bits of the two blocks are filled with P_Z and P_S individually. Following the procedure above, we can obtain the plaintext block P_j.

In order to demonstrate the security loophole caused by the flaw, firstly, we encrypt a plaintext sequences, "abcdefghijklnmopqrstuvwxyz", via the key $(\mu, x_0) = (3.999995, 0.17777)$. We fill the first 2 blocks of plaintext P_j, D_j, A_j, and the corresponding ciphertext C_j into Table 1.

Encrypt the special plaintexts P_Z and P_S. Table 2 shows the result.

Recover the first block of the ciphertext: at first, deal with the ciphertext blocks C_Z and C_S which correspond to the plaintext blocks P_Z and P_S. We can obtain A_1 and D_1. Put them into Eqs. (9) and (11), we have

$$C_Z = A_1 = 0 \times 5d55a4bc5a8ae20f,$$
$$C_S \oplus C_Z = 0 \times \text{fffffffeffffffff}.$$

After analyzing the values, we know the $D_1 = 0 \times 1f$. With the help of A_1 and D_1, we can obtain the plaintext block "abcdefgh". From Eq. (5), we know $f(C_1) \bmod 64 = 0 \times 00000017$. We exhaust the permutation $P_{D_1^*}$, trying to find out a block $P_{D_1^*}$ which can satisfy the Eq. (13). Table 3 shows the procedure.

Recover the second block of the ciphertext: we find out that when $P_{D_1^*} = 0 \times 0800000000000000$, the Eq. (13) is satisfied. We also obtain the C_1' which corresponds to the $P_{D_1^*}$. Construct two new sequences P_{Z2} and P_{S2} whose lengths are 2 blocks. Make sure that the first block of P_{Z2} and P_{S2} are filled with C_1' individually and the second block of P_{Z2} and P_{S2} are filled with P_Z and P_S individually. Table 4 shows the details of the operations above. Consider C_{Z2} and C_{S2}, the second block of the ciphertext after encrypting the 2 sequences. We have

$$C_{Z2} = A_2 = 0 \times 1a53363e031eb2b3,$$
$$C_{S2} \oplus C_{Z2} = 0 \times \text{ffffffffbfffff}.$$

We obtain $D_2 = 0 \times 29$ after analyzing the 2 values above. With the help of A_2 and D_2, we can recover the second block of the plaintext, "ijklmnop". Deal with the second block of P_{Z2}, and make it satisfy Eq. (14). Following the method of attacking the second block of the ciphertext, we can recover all the plaintext.

Table 4
Encryption of the plaintext $P_{Z2}$and P_{S2}.

j	P_{Z2}	C_{Z2}	D_j	A_j	P_{S2}	C_{S2}
1	0×0800000000000000	0×5d55a4bc4a8ae20f	0×1f	0×5d55a4bc5a8ae20f	0×0800000000000000	0×5d55a4bc4a8ae20f
2	0×0000000000000000	0×1a53363e031eb2b3	0×29	0×1a53363e031eb2b3	0×7fffffffffffff	0×e5acc9c1fca14d4c

Table 5
Second block of the encryption of the plaintext "abcdefghijklnmopqrstuvwxyz" with modified cryptosystem using $(\mu, x_0) = (3.999995, 0.17777)$.

j	P_j	D_j	A_j	C_j
1	0×6162636465666768	0×1f	0×5d55a4bc5a8ae20f	0×6dc9ed3375b836d6
2	0×696a6b6c6d6e6f70	0×29	0×1a53363e031eb2b3	0×9e8ea33dc0488073
3	–	–	–	–

Table 6
Encryption of the plaintext P_Z and P_S with modified cryptosystem using $(\mu, x_0) = (3.999995, 0.17777)$.

	PlainText	D_1	A_1	*Cipher*
P_Z	0×0000000000000000	0×1f	0×5d55a4bc5a8ae20f	0×b7f8815e8eddf275
P_S	0×7fffffffffffff	0×1f	0×5d55a4bc5a8ae20f	0×48077ea171320d8a

Attack algorithm 2 is similar to attack algorithm 1, except the method of constructing new sequences. In attack algorithm 2, the new sequence consists of all the recovered plaintext blocks and a special block. Using the recovered blocks can ensure that the special block is encrypted by the same key. With this guarantee, we can analyze the corresponding A_j and D_j.

3. Analysis of the flaw and remedial improvement

Attack algorithm 1 adoptsa brute-force method to satisfy. Eq. (13). It means that we need to exhaust 2^{64} permutations because the length of each block is 64. So is the key. Due to the Eqs. (5) and (6), the complication is sharply reduced. The space reduces from 2^{64} to 64. Based on the theory of finite groups, the XOR operation is the simplest and most effective way to permute the order of the element in a group. After processing the Eqs. (5) and (6), the solution space is divided into 64 partitions with equal distribution. You can find out the values you need with less than 64 times computation. If we want to attack the ciphertexts, we only need to guarantee that the keystream we obtain is the same as the encrypting one.

The cryptosystem proposed by [4] has some excellent benefits. It combines a chaotic map, block encryption and stream encryption with less computation. But the two fatal flaws make it not suitable to be applied widely for its security. Although the keystreams depend on plaintext, the first key of the keystreams does not. According to this flaw, the cryptanalyst can obtain the key from the first block. The other flaw is that the ciphertext has too much information, and the algorithm relies directly on ciphertext. Therefore, the cryptanalyst can obtain the keys block by block, so it is easy to synchronize the track of the chaotic map.

This paper proposes an improvement to remedy the flaws, via hiding D^*, decreasing the direct dependency between algorithm and ciphertext. After step 5 in encryption, add Eq. (19). Meanwhile, in order to avoid the possibility of guessing the original plaintext by trial shift and XOR using P_Z and P_S, we add functions $f_1(x)$ and $f_2(x)$ to make the ciphertext shift uncertain bits. Now, the procedure of encryption can be described as follow:

$$f_1 = (A_j + D_j) \bmod 64, \tag{15}$$

$$C'_j = F(C_j, f_1(D_j)) \oplus A_j, \tag{16}$$

$$D^* = D_j + f(C'_j) \bmod 64, \tag{17}$$

$$f_2 = (A_j + 2D_j) \bmod 64, \tag{18}$$

$$C''_j = F(C'_j, f_2(D_j)) \oplus A_j. \tag{19}$$

Hence we can obtain D^* directly from ciphertext. Though this improvement adds some computations, they are not time-consuming operations. So this algorithm does not lose the original advantage. This improvement fixes the 2 flaws of the algorithm and enhances the security of the cryptosystem. When we decrypt the ciphertext, we can see that D^* only exists in the decryption process.

Obviously, the direct dependency between algorithm and ciphertext has been eliminated. Tables 5 and 6 illustrate the efficiency of this remedy.

4. Conclusion

According to Refs. [3,4], two important things should be considered when designing a chaotic cryptosystem. The first one is to avoid the same secure key to generate the unchangeable keystream. The second one is to avoid using the algorithm

which relies on the ciphertext directly, because it may help a cryptanalyst to trace the track of the chaotic dynamic system. This paper analyzes the cryptosystem proposed by [4], and points out the 2 fatal flaws in it, finally remedies the flaws of algorithm without losing the original benefits, and enhances the security of the cryptosystem.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No: 60573172), the Superior University doctor subject special scientific research foundation of China (No: 20070141014) and the National Natural Science Foundation of Liaoning province (No: 20082165).

References

[1] S. Behnia, A. Akhshani, S. Ahadpour, H. Mahmodi, A. Akhavan, A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps, Phys. Lett. A 366 (4–5) (2007) 391.
[2] M.S. Baptista, Cryptography with chaos, Phys. Lett. A 240 (1–2) (1998) 50.
[3] T. Xiang, X. Liao, G. Tang, Y. Chen, K.W. Wong, A novel block cryptosystem based on iterating a chaotic map, Phys. Lett. A 349 (1–4) (2006) 109.
[4] Y. Wang, X. Liao, T. Xiang, K.W. Wong, D. Yang, Cryptanalysis and improvement on a block cryptosystem based on iteration a chaotic map, Phys. Lett. A 363 (4) (2007) 277.
[5] K.W. Wong, S.W. Ho, C.K. Yung, A chaotic cryptography scheme for generating short ciphertext, Phys. Lett. A 310 (1) (2003) 67.
[6] G. Álvarez, F. Montoya, M. Romera, G. Pastor, Cryptanalysis of an ergodic chaotic cipher, Phys. Lett. A 311 (2–3) (2003) 172.
[7] A. Palacios, H. Juarez, Cryptography with cycling chaos, Phys. Lett. A 303 (5–6) (2002) 345.
[8] D.R. Stinson, Cryptography: Theory and Practice, CRC Press, Boca Raton, FL, 1995.
[9] W. Stallings, Cryptography and Network Security: Principle and Practice, Pearson Education, US, 2006.